



Progettazione del Software

Analisi: UML Use Cases & Documenti di Specifica

Domenico Fabio Savo

Dipartimento di Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti

SAPIENZA Università di Roma

Le slide di questo corso sono il frutto di una rielaborazione di analogo materiale redatto da Marco Cadoli, Giuseppe De Giacomo, Maurizio Lenzerini e Domenico Lembo



Il diagramma degli use case

Ricordiamo che lo schema concettuale (per le finalità di questo corso) è costituito da:

- Diagramma delle classi (e degli oggetti)
- Diagramma degli use case.

Il **diagramma degli use-case** descrive le funzionalità fondamentali che il sistema deve realizzare, in termini di scenari di utilizzo del sistema.



Use case (1)

Un **use case** rappresenta una tipica interazione tra un **utente** ed il sistema software da realizzare.

Un use case cattura una qualche funzione visibile dall'utente, e la sua descrizione si ottiene attraverso l'interazione tra analista ed utente in fase di analisi.

In altre parole, un use case definisce un particolare modo di utilizzare il sistema, il quale offre **servizi** e **funzionalità** in risposta a eventi prodotti da attori esterni.



Use case (2)

Un use case modella un processo (o un insieme di processi) che è **trasversale** rispetto alle classi, cioè **coinvolge più classi allo stesso livello**, e sarebbe una forzatura modellarlo come una operazione di una singola classe.

Un use case è in genere composto da diverse **operazioni**, che non vengono definite in modo dettagliato nel diagramma.

Vedremo, quando parleremo di “specificazione di un use case”, come queste operazioni vengono definite.



Use Case: Attori

Un **use case** è formulato sulla base delle **funzionalità** offerte dal sistema così come sono percepite dagli utenti.

Oltre agli use case, un altro componente fondamentale del diagramma degli use case è l'attore. Un **attore** è un ruolo che un utente (una persona o un sistema esterno) riveste interagendo con il sistema.

- Lo stesso utente può essere rappresentato da più di un attore, cioè lo stesso utente può rivestire più ruoli.
- Più utenti possono essere rappresentati dallo stesso attore.



Use case: diagramma

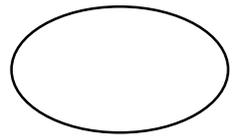
Un **diagramma degli use case** è un **grafo** i cui **nodi** possono essere

- Attori
- Use Case

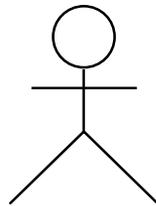
mentre gli **archi** rappresentano

- la **comunicazione** tra gli attori e gli use case
- i **legami d'uso** tra use case
- l'**estensione** di uno use case da parte di un altro

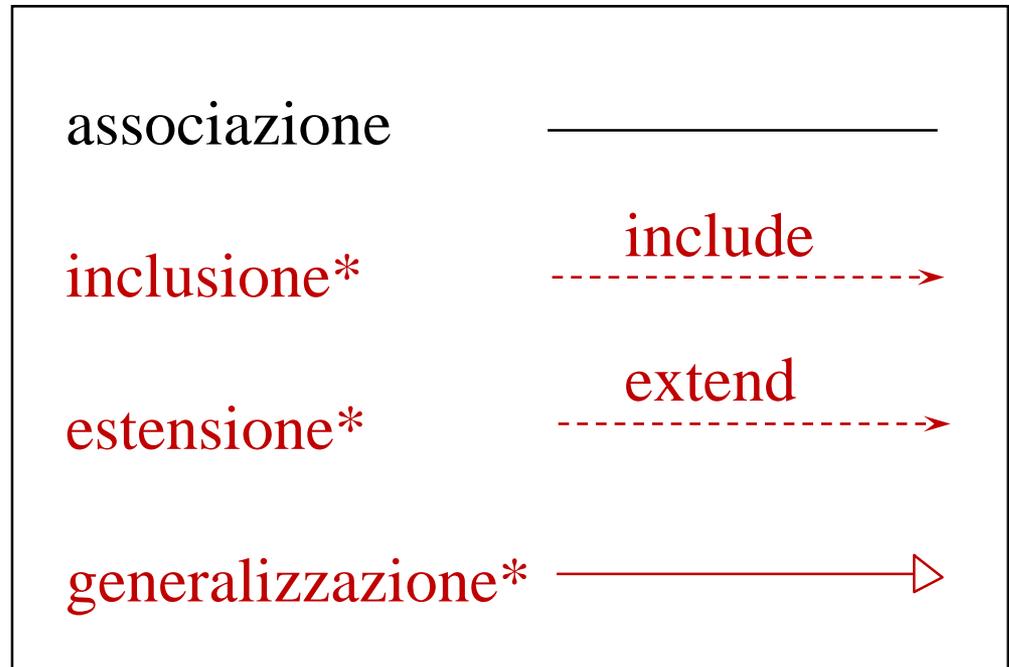
Componenti di un diagramma degli use case



Use Case



Attore



Relazioni principali nel diagramma degli use case

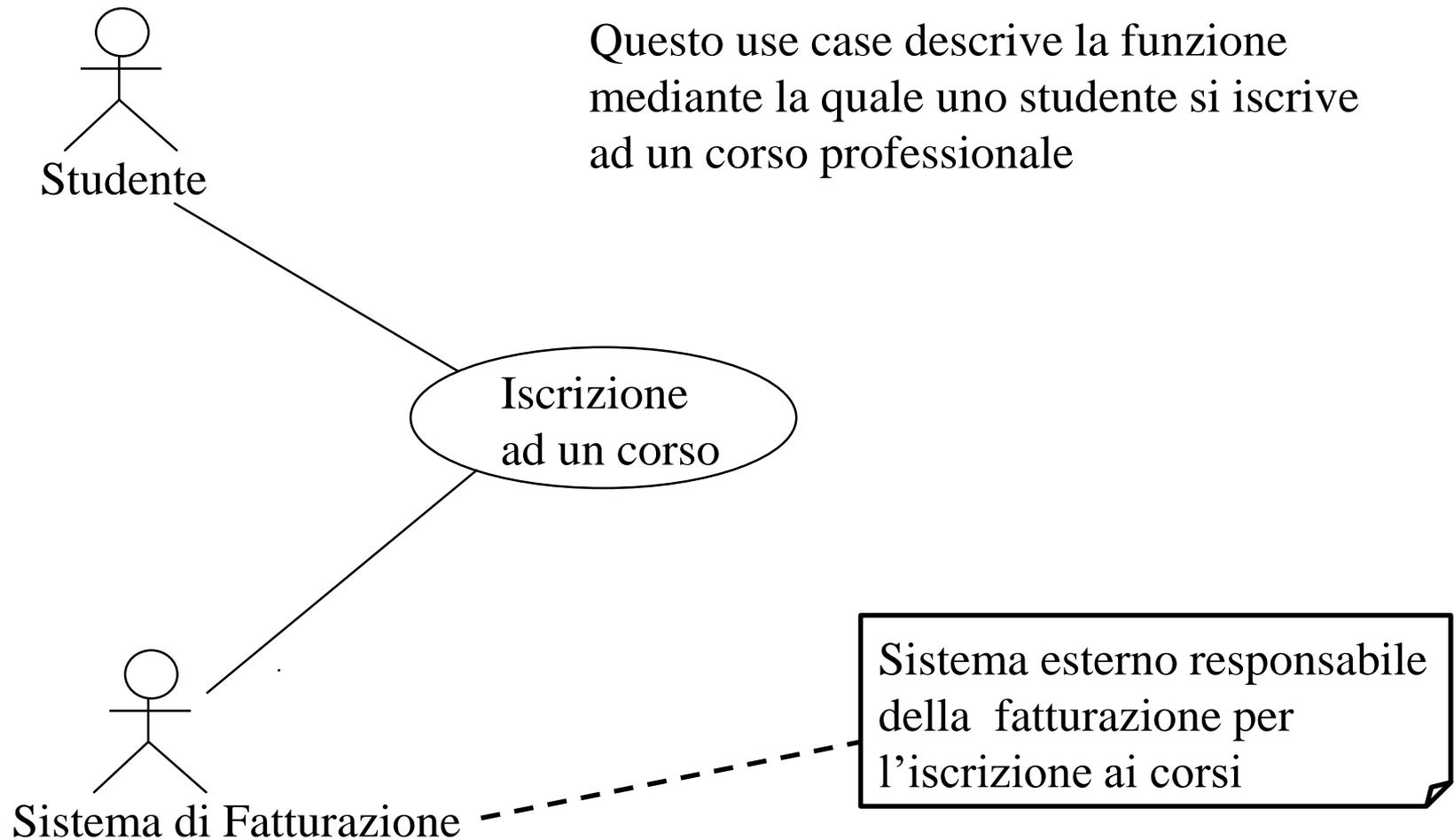
*Non considerate in questo corso

Associazione

La partecipazione di un attore ad uno use case è rappresentata da un arco di **associazione** tra il simbolo dell'attore e il simbolo di use case. In generale, **questo significa che l'attore “comunica” con lo use case, e cioè può eseguire le funzionalità dello use case.**



Esempio di diagramma degli use case





La specifica

Lo schema concettuale viene alla fine corredato da

- una **specifica** per ogni **Classe**
- una **specifica** per ogni **Use case**

La **specifica di una classe** ha lo scopo di definire precisamente il **comportamento di ogni operazione della classe**

La **specifica di un use case** ha lo scopo di definire precisamente il **comportamento di ogni operazione di cui lo use case è costituito**

Specifica di una classe

La specifica di una classe C ha la seguente forma:

InizioSpecificaClasse C

Specifica della operazione 1

...

Specifica della operazione N

FineSpecifica

Specifica di un Use Case

La specifica di un use case si fornisce facendo la lista delle operazioni (una o più) che costituiscono lo use case stesso, e fornendo poi la specifica di ogni operazione.

La specifica di un use case D ha la seguente forma:

InizioSpecificaUseCase D

Specifica della operazione 1

...

Specifica della operazione N

FineSpecifica

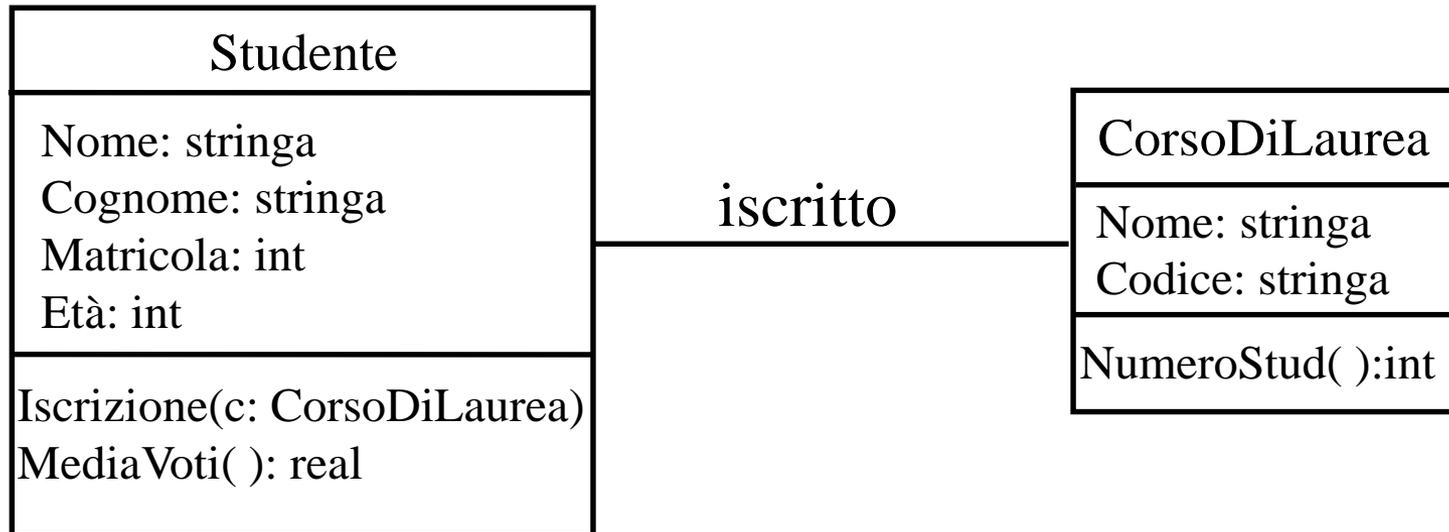
Specifica di una operazione

Che sia una operazione di una classe o una operazione di un use case, la specifica di una operazione ha la seguente forma:

alfa (X1: T1, ... , Xn: Tn): T
pre: *condizione*
post: *condizione*

- **alfa (X1: T1, ... , Xn: Tn): T** è la **segnatura** dell'operazione (T può mancare),
- **pre** rappresenta la **precondizione** dell'operazione, cioè l'insieme delle condizioni (a parte quelle già stabilite dalla segnatura) che devono valere **prima** di ogni esecuzione della operazione
- **post** rappresenta le **postcondizioni** della operazione, cioè l'insieme delle condizioni che devono valere **alla fine** di ogni esecuzione della operazione

Esempio di specifica di una operazione



InizioSpecificaClasse CorsoDiLaurea

NumeroStud() : int

pre : nessuna

post : **result**   uguale al numero di studenti iscritti nel
corso di laurea **this**

FineSpecifica

Precondizioni e postcondizioni (1)

Nella specifica di una operazione, nella **precondizione** si usa

- “**this**” per riferirsi all’oggetto di invocazione della operazione

Nella specifica di una operazione, nella **postcondizione** si usa

- “**this**” per riferirsi all’**oggetto di invocazione** della operazione nello stato corrispondente **alla fine** della esecuzione della operazione
- “**result**” per riferirsi al **risultato** restituito dalla esecuzione della operazione
- **pre(alfa)** per riferirsi al valore della espressione **alfa nello stato corrispondente alla precondizione**

Precondizioni e postcondizioni (2)

Precondizioni e Postcondizioni specificano in maniera **dichiarativa** le condizioni (*il cosa*), e non indicano il modo con cui si verificano tali condizioni (*il come*)

Nella specifica si fa in genere riferimento alle classi ed alle associazioni del class diagram, ai parametri in input e, in caso i parametri siano oggetti, alle loro proprietà (se necessario anche dinamiche – operazioni), accedute tramite la notazione ‘.’

Precondizioni e postcondizioni (3)

- *Nella specifica delle operazioni di classi*, si può fare anche riferimento all'oggetto di invocazione (tramite la parola chiave **this**) ed alle sue proprietà
- *Nella specifica di uno use case* è possibile fare riferimento ad altri use case. Non ha invece senso parlare di oggetto di invocazione (e quindi non si può usare **this**)
- La specifica può essere data in linguaggio naturale oppure in maniera formale, con notazione matematica. Diversi livelli di precisione sono possibili (in seguito si forniranno versioni alternative). Qualunque sia il livello di precisione scelto, la specifica deve mantenere il carattere dichiarativo (non ci devono essere dettagli procedurali, ad es. parti di algoritmi). Utilizzare i nomi delle classi e delle associazioni del class diagram e fare riferimento la sua semantica è un modo per definire specifiche corrette e significative.

Esempio

*Si vogliono modellare gli studenti (con matricola, età e **numero di esami***), il corso di laurea in cui sono iscritti, ed i corsi di cui hanno sostenuto l'esame, con il professore che ha verbalizzato l'esame, ed il voto conseguito. Di ogni corso di laurea interessa il codice e il nome. Di ogni corso interessa il nome e la disciplina a cui appartiene (ad esempio: matematica, fisica, informatica, ecc.). Di ogni professore interessa codice, età e **numero di verbalizzazioni*** effettuate. Si noti che uno studente non può sostenere più di un esame per lo stesso corso.*

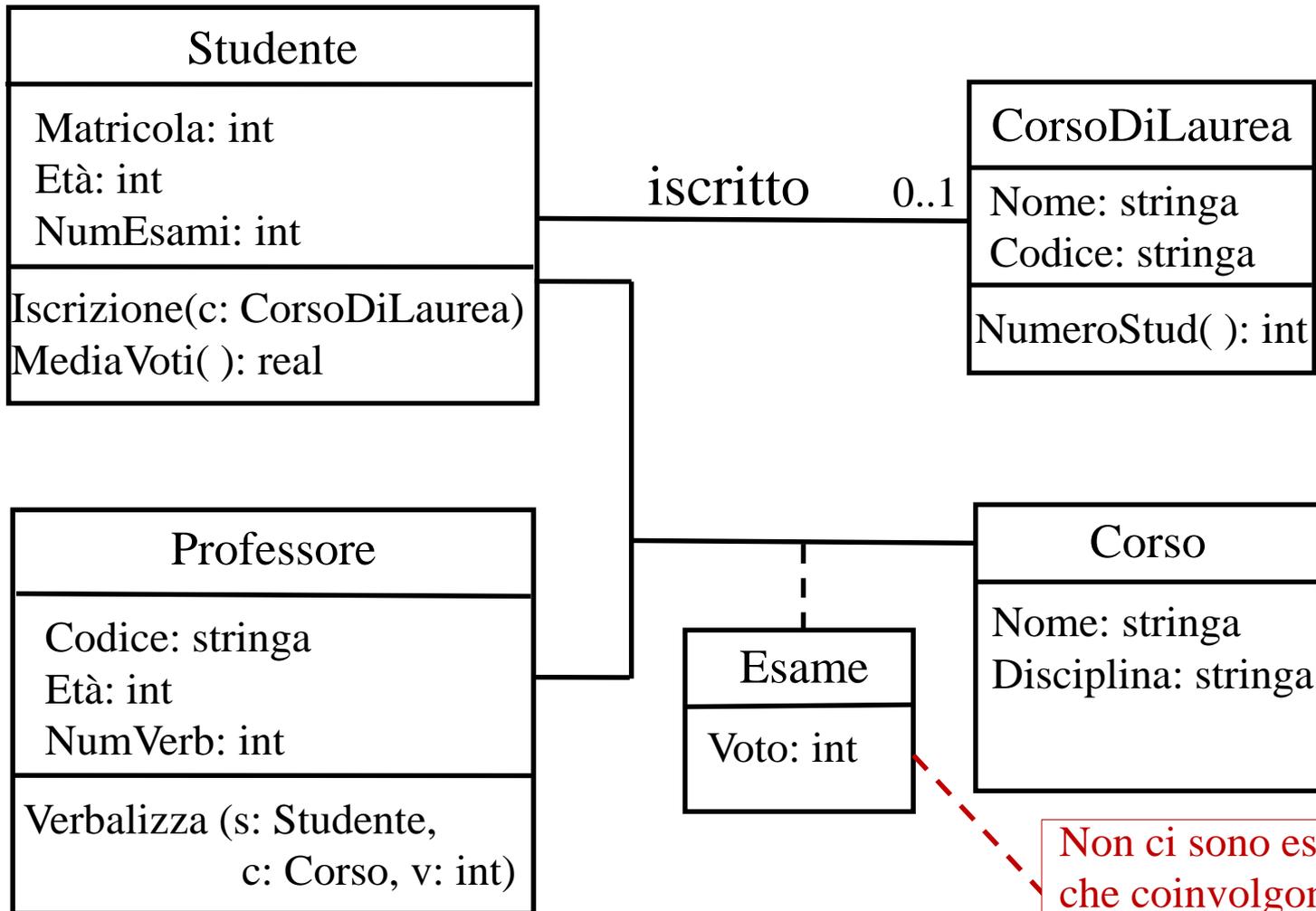
Al momento dell'iscrizione, lo studente specifica il corso di laurea a cui si iscrive.

Dopo l'effettuazione di un esame, il professore comunica l'avvenuta verbalizzazione dell'esame con i dati relativi (studente, corso, voto).

La segreteria vuole periodicamente calcolare la media dei voti di uno studenti e il numero di student iscritti ad un corso di laurea.

**numero di esami e numero di verbalizzazioni per semplicità saranno modellati come attributi. E' possibile (e ragionevole) modellarli come operazioni (lasciato come ulteriore esercizio)*

Esempio di diagramma delle classi



Non ci sono esami diversi che coinvolgono lo stesso studente e lo stesso corso

Esempio di specifica di classi (1)

InizioSpecificaClasse Professore

Verbalizza(s: *Studente*, c: *Corso*, v: *int*)

pre: s non ha ancora sostenuto l'esame c (*non esiste un link di tipo Esame che colleghi s, c ed un qualsiasi professore*), e $18 \leq v \leq 31$

post: s ha sostenuto l'esame per il corso c con il professore this con voto v (*this, s e c sono collegati da un link di tipo Esame, con voto v*). Inoltre vale che $s.\text{NumEsami} = \text{pre}(s.\text{NumEsami}) + 1$, e $\text{this}.\text{NumVerb} = \text{pre}(\text{this}.\text{NumVerb}) + 1$

FineSpecifica

Nota: *Fra parentesi, in corsivo, è indicata una versione alternativa, con precisi riferimenti al class diagram ed alla sua semantica (istanze, link, ecc)*

Esempio di specifica di classi (2)

InizioSpecificaClasse Studente

Iscrizione(c: CorsoDiLaurea)

pre: this non è iscritto ad alcun corso di laurea (*non esiste alcun link di tipo iscritto che coinvolge this*)

post: this è iscritto al CorsoDiLaurea c (*this e c sono collegati da un link di tipo iscritto*)

MediaVoti() : real

pre: this è iscritto ad un corso di laurea (*esiste un link di tipo iscritto che coinvolge this*) e this.NumEsami > 0

post : result è la media dei voti degli esami sostenuti da this

FineSpecifica

Esempio di specifica di classi: notazione formale

InizioSpecificaClasse Professore

Verbalizza(s: Studente, c: Corso, v: int)

pre : $\neg(\exists p \mid p \in \text{Professore} \wedge \langle s, p, c \rangle \in \text{Esame})$
 $\wedge 18 \leq v \leq 31$

post : $\text{Esame} = \text{pre}(\text{Esame}) \cup \{ \langle s, \text{this}, c \rangle \} \wedge$
 $\text{Esame.voto}(\langle s, \text{this}, c \rangle) = v \wedge$
 $s.\text{NumEsami} = \text{pre}(s.\text{NumEsami}) + 1 \wedge$
 $\text{this.NumVerb} = \text{pre}(\text{this.NumVerb}) + 1$

FineSpecifica

- \neg , \wedge , \exists rappresentano rispettivamente il *NOT*, l'*AND*, e l'*Esiste*.
- Si noti che **Esame** rappresenta **tutte le istanze dell'associazione** Esame (un insieme di terne di oggetti studente-professore-corso)
- **Un'istanza di associazione si rappresenta con le parentesi angolari**, che racchiudono l'elenco degli oggetti che la costituiscono (es., $\langle s, p, c \rangle$)
- Il valore che un attributo **F** di un'associazione **A** assegna ad un'istanza $i \in A$ si denota con **A.F(i)** (ad es., **Esame.voto($\langle s, \text{this}, c \rangle$)**)

Specifica mediante una notazione formale

InizioSpecificaClasse Studente

Iscrizione(c: CorsoDiLaurea)

pre : $\neg(\exists c2 \mid c2 \in \text{CorsoDiLaurea} \wedge \langle \text{this}, c2 \rangle \in \text{iscritto})$

post : $\text{iscritto} = \text{pre}(\text{iscritto}) \cup \{ \langle \text{this}, c \rangle \}$

MediaVoti() : real

pre : $(\exists c \mid c \in \text{CorsoDiLaurea} \wedge \langle \text{this}, c \rangle \in \text{iscritto}) \wedge \text{this.NumEsami} > 0$

post : definiamo Voti come l'insieme

$\{ \langle c, v \rangle \mid \exists p \mid p \in \text{Professore} \wedge c \in \text{Corso} \wedge$
 $\langle \text{this}, p, c \rangle \in \text{Esame} \wedge \text{Esame.voto}(\langle \text{this}, p, c \rangle) = v \}$

$$\text{result} = \frac{\sum_{\langle c, v \rangle \in \text{Voti}} v}{\text{this.NumEsami}}$$

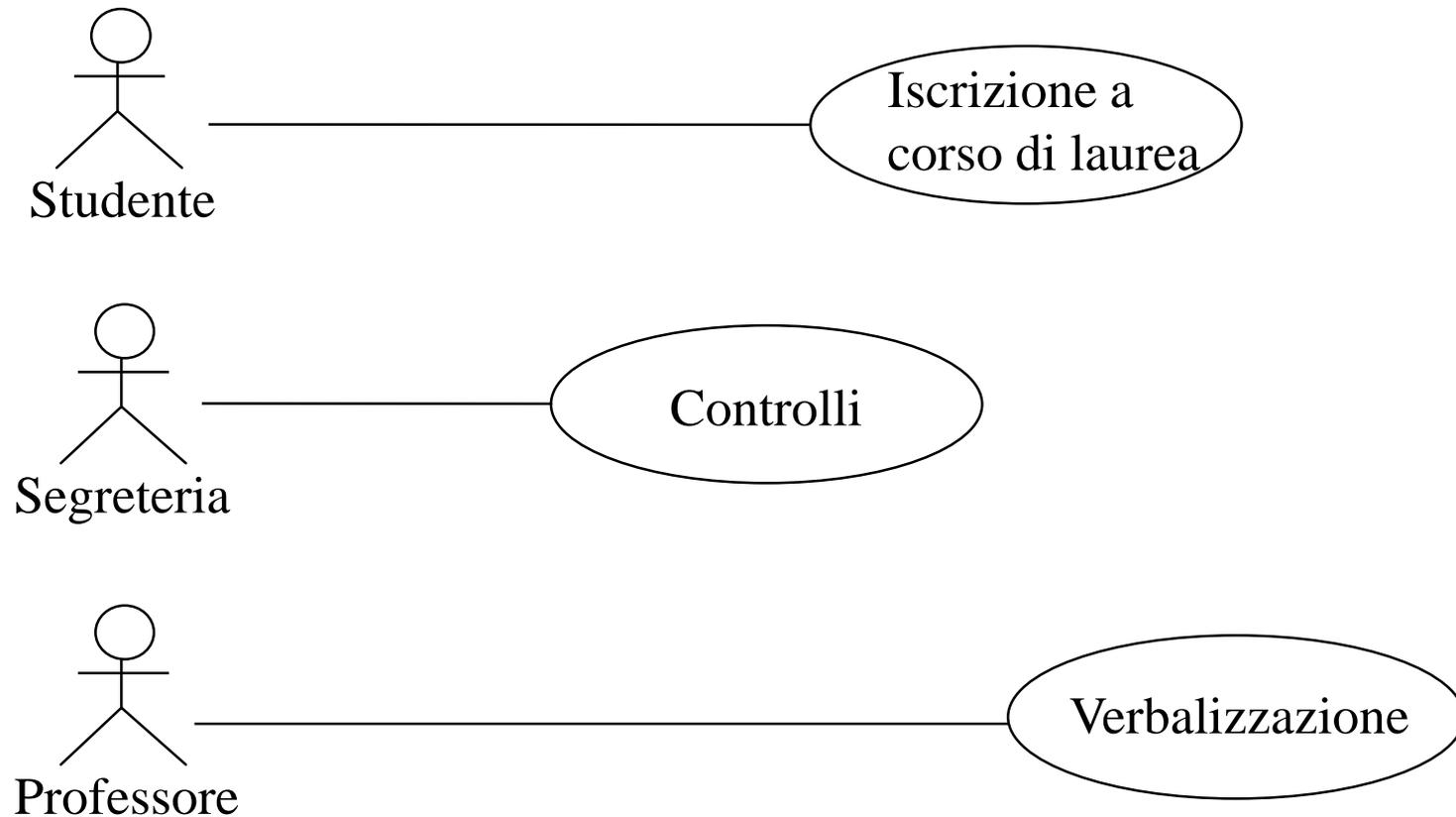
FineSpecifica

Nota: Definire Voti come l'insieme

$\{ v \mid \exists p \mid p \in \text{Professore} \wedge c \in \text{Corso} \wedge \langle \text{this}, p, c \rangle \in \text{Esame} \wedge \text{Esame.voto}(\langle \text{this}, p, c \rangle) = v \}$

Sarebbe stato scorretto: voti uguali in esami diversi sarebbero contati una sola volta!

Esempio di diagramma degli use case



Esempio di specifica di use case

InizioSpecificaUseCase Controlli

MediaVoti(s: Studente): real

pre : s è iscritto ad un corso di laurea e s.NumEsami > 0

post : result = s.MediaVoti() // invoca operazione di Studente

NumeroStudenti(c: CorsoDiLaurea): int

pre : nessuna

post : result = c.NumeroStud() //invoca operazione di
CorsoDiLaurea

FineSpecifica

Specifica mediante una notazione formale

InizioSpecificaUseCase Controlli

MediaVoti(s: Studente): real

pre : $(\exists c \mid c \in \text{CorsoDiLaurea} \wedge \langle s, c \rangle \in \text{iscritto}) \wedge$
s.NumEsami > 0

post : result = s.MediaVoti()

NumeroStudenti(c: CorsoDiLaurea): int

pre : true

post : result = c.NumeroStud()

FineSpecifica

Esempio

- Nel caso si richieda di avere in input o come output di una operazione un insieme di oggetti istanze di una classe A (o di valori istanze di un tipo di dato D), si può utilizzare il costrutto $Insieme(A)$ (resp. $Insieme(D)$).
- Ad esempio, per l'operazione di caso d'uso: **Dato un insieme di studenti si vuole conoscere l'insieme di corsi di laurea a cui sono iscritti.**
- ...abbiamo la seguente specifica

$CdLPerStudenti(Insieme(Studente)):Insieme(CorsoDiLaurea)$

pre : nessuna

post : result è l'insieme di istanze di $CorsoDiLaurea$ coinvolte in almeno un link di tipo $iscritto$ con almeno un elemento di $Insieme(Studente)$



Esercizio 19 (continua l'esercizio 15)

Di ogni officina interessano il nome, l'indirizzo, il numero di dipendenti, i dipendenti (con l'informazione su quanti anni di servizio ogni dipendente ha svolto presso l'officina), ed il direttore (unico). Un dipendente lavora presso una ed una sola officina, ed un direttore dirige una ed una sola officina. Il direttore non è un dipendente. Di dipendenti e direttori interessano: il codice fiscale (CF), l'indirizzo ed il numero telefono. Dei direttori interessa anche l'età. Delle riparazione interessa: il codice, l'officina presso cui è eseguita, il veicolo riparato, l'ora e la data di accettazione, e, per le riparazioni portate a termine, data e ora di riconsegna. Di ogni veicolo interessano: il modello, il tipo, la targa, il proprietario e l'anno di immatricolazione. Dei proprietari interessa il CF, l'indirizzo, ed il numero telefono. Data una officina, gli utenti del sistema vogliono conoscere quali sono i suoi dipendenti, chi è il direttore, quali sono le riparazioni effettuate. Inoltre per ogni direttore è necessario poter risalire alla officina diretta. Infine, dato un veicolo si vuole conoscere il suo proprietario, e data una riparazione si vogliono conoscere il veicolo ed il proprietario.

Completare il Class Diagram precedente aggiungendo le operazioni di interesse e fornire la specifica della classe Officina, della classe Riparazione e della classe Direttore.

Inoltre, tracciare il diagramma degli Use Case e fornire la specifica di ogni Use Case



Esercizio 19 (soluzione)



Esercizio 19 (soluzione)



Esercizio 19 (soluzione)



Esercizio 19 (soluzione)



Esercizio 19 (soluzione)



Esercizio 19 (soluzione)



Esercizio 19 (soluzione)



Esercizio 19 (soluzione)